

United States Patent Application

Title

DOCUMENT PROCESSING METHOD AND SYSTEM, AND
COMPUTER-READABLE RECORDING MEDIUM HAVING
DOCUMENT PROCESSING PROGRAM RECORDED THEREIN

Inventors

Yuki AOYAMA,
Yukie TAKITA,
Toru TAKAHASHI,
Yukio HOSHI.

BACKGROUND OF THE INVENTION

The present invention relates to techniques for creating and processing a document with use of a word processor or personal computer for its display and print and more particularly, to a technique for processing a document which uses information that frequently appears in the document as shared information.

For the purpose of efficiently sharing and re-using document information, a recent active move is to use such a document language as a standard generalized markup language (SGML) or XML to obtain and utilize a document.

Since a document prepared in a structured document language can be prepared by dividing the document into its structural elements as document parts for editing and describing links to the respective parts, it is easy for the document parts to be shared with another document or to re-use another document on every part basis.

When it is desired to display or print the link-described document having such links to the parts described therein, the document converted to a document having the contents of the document parts of the other link-destination document embedded in the locations, and then displayed or printed.

To this end, a conventional editor for editing such a document provides a means for designating a document part as a part of the other document to describe a link to the document in the editing document.

5 When such a document editor is used, it can facilitate such an operation that a figure, which was prepared in a functional specification as another document, can be re-used, for example, during editing of a design document.

10 Further, when a re-used part was edited, its edited result can be reflected throughout the entire document having the shared part. As a result, consistency management can be facilitated.

 JP-A-8-44718 (Fuji Xerox Ltd.) discloses a
15 method that, when a figure or table is shared among a plurality of documents, its shared relationships are managed so that, even the shared constituent element is edited in the sharing original document, its edited result can be correctly reflected even on the other
20 shared documents (figure or table), thus maintaining a consistency of the shared information.

 In addition, even in a document system for performing edition management, the above method can keep the shared information consistency. Thus, when it is
25 desired to reduce the number of versions, it becomes possible to judge whether a constituent element or elements are shared among a plurality of versions, thus realizing such a management as to prevent an

inconsistency in the shared relationships.

Constituent elements of documents which are effective when the elements are shared include, in addition to such a figure or table as mentioned in connection with JP-A-8-44718, a character string which
5 are used in many documents.

For example, at the time of preparing a document, it is very often for a document writer to consistently use the same character string as in a
10 plurality of documents or in one document. More specifically, when it is desired to prepare specifications in a product development project, it is required to use common module names, unique abbreviations and coinages which constitute the project.

15 In reality, however, it is often required to modify such words to more appropriate ones as the project goes on. In such a case, it has been conventionally inevitably required to manually modify all the documents at the modified time point.

20 When such documents are written as shared by a plurality of writers, it becomes further difficult to obtain a word consistency throughout the documents and thus it inevitably involves an enormous amount of labor of finally proof-reading the documents.

25 Further, person's names or belonging departments are frequently appear in a plurality of documents regardless of the document types. For this reason, when a belonging department was modified, it is also required

to modify many associated documents.

Furthermore, when a document is prepared, it is often not to prepare the document newly but to copy the same type of document as a template and to edit the contents thereof. Thus, even when it is required to change the old department to the new one, it is often to erroneously leave some of the old departments as they are without being modified to the new ones.

When such modification information is treated as a document part to be shared among a plurality of documents, the information consistency can be automatically realized. And even when the information is corrected, its corrected contents can be reflected on all the documents. As a result, highly efficient document preparation and editing can be carried out.

However, when such shared information is edited in the conventional system for its sharing, this involves problems which follow.

Fig. 5 shows an example of a plurality of documents which contain a document part or parts as a part or parts of the other documents with use of a known document editor.

For example, when a figure 504 (which is called share originator information), which forms one of constituent elements of a document A 501, is designated to describe its link to a document B 502, this causes appearance of contents of the document part (figure 504) of a link destination in the document B 502 (as a figure

505).

Next, when it is desired to share the same figure 504 even in a document D 506, a document containing the share originator information, that is, the document A 501 is searched to perform similar share designation to the above.

When information is shared in such a large unit as the figure and the number of sharing units as document parts is not so large as in the example of Fig. 5, the conventional system may be applicable. However, when such document parts such as terms, names and belonging departments appear frequently, this share designating operation becomes highly troublesome.

That is, when many pieces of information are to be shared in such a small unit, it becomes difficult in the share designation to search the other documents containing the share originator information. Meanwhile, it is often that an identical type of shared information is simultaneously modified as when many terms are modified or a plurality of persons changes their belonging departments at a time in a project. In this case, the prior art system is required to search and modify the other documents containing the share originator information, which makes the modifying operation complex.

Meanwhile, information on person's name or belonging department is associated with each other, and thus when a person is determined, its name, belonging

department, extension telephone number ought to be automatically determined. However, when the prior art system is employed to edit the document to share information associated with the person, its name, 5 belonging department and extension telephone number are shared respectively independently. For this reason, modification of, e.g., name cannot follow automatic modification of the belonging department and extension telephone number associated with the name and require 10 individual editing of the shared information. As a result, its editing operation becomes highly complicated.

For the purpose of controlling such related information, a database management system is employed. 15 This system however has a problem that, since schema design for determining a data structure is difficult in database, the system cannot be easily applied to document preparation. Another problem is that, once the data structure is determined, it becomes hard to modify 20 the structure.

Further, when the shared information was modified, there are considered two cases where it is desired to reflect the modification on all documents having the shared part and the other, it is desired not 25 to reflect the modification on past documents already edited.

For example, in the case of a term used in a project, it is necessary to reflect the term modifi-

cation even on the documents already edited, but with respect to reporter's name and belonging department described in a report submitted in the past, the information at the time of the submission must remain.

5 The prior art system has a problem that the management of consistency between the above two cases cannot be realized since the part modification is inevitably reflected on all the documents having the shared part.

10 The prior art system also has another problem that, when it is desired to modify shared information and to reflect the information modification on documents having the shared information, the correlation between the document and shared information link varies
15 depending on the modification contents of the shared information, resulting in a wrong reflection thereof.

SUMMARY OF THE INVENTION

In view of such problems in the prior art, it is therefore an object of the present invention to
20 facilitate unification of notation or representation of information such as term, name and belonging department and also to facilitate automatic reflection of modification on all documents. A specific object of the present invention is to provide a document processing
25 method and system, and a recording medium having a document processing program, which enables easy description of a link of shared information in a

document.

Another object of the present invention is to provide a document processing technique which, unlike such a complex method as in a database management
5 system, can facilitate data structure modification in a simple manner, and thus which can manage related information and efficiently edit documents with use of the related information.

A further object of the present invention is
10 to provide a document processing technique, for shared information having been modified, which can manage the consistency between two cases where it is desired to reflect a modification result on all documents having its shared information part and where it is desired not
15 to reflect the modification on past documents already edited.

Yet another object of the present invention is to provide document processing technique which can avoid variations in correlations between documents and shared
20 information link, even when the shared information is modified and it is desired to reflect its modification result on documents having the shared information.

To achieve the above objects, in accordance with an aspect of the present invention, there is
25 provided a computer-implemented method for processing a document in a computer implementation manner in a document processing system having a processor, a memory and a terminal device, comprising the steps of:

generating as a set of shared information a shared
information block frequently appearing in a plurality of
documents; storing the generated shared information in a
memory; listing up the stored shared information for
5 each of information types; editing the document to
describe a link to the shared information selected from
the displayed list of the edited shared information in
the documents; storing the link-described documents in
the memory; and selecting and reading out from the
10 memory the generated shared information and described
document, embedding the shared information in the link-
described document according to the links described
therein and the generated shared information, and
outputting information indicative of a document having
15 the shared information embedded therein. The output
step may include a step of displaying or printing the
output information.

In accordance with another aspect of the
present invention, the information block generating step
20 generates as the shared information a term such as
abbreviation or coinage and a term combination of a term
such as person's name or belonging department and edits
the shared information into structured documents
depending on the types of the information.

25 In accordance with a further aspect of the
present invention, the document processing method
further comprises a version management step of managing
a change history of the shared information and

documents, and the version management step stores in the memories the generated shared information and the generated document attached by information indicative of a new version, generates version information for each of
5 the shared information and document and stores the version information in the memory, and the output step includes a step of displaying or printing the document having the shared information embedded therein on the basis of the output information. The display or print
10 step, when it is desired to display or print the document in the display or print step, selects the shared information of the latest version created prior to a date and hour of creation of the document on the basis of the version information and embeds the selected
15 shared information in the document.

In accordance with yet another object of the present invention, the document processing method further comprises a step of managing a change history of the shared information and document, the version
20 management step stores in the memory the generated shared information and generated document attached by information as a new version, creates version information for each of the shared information and document and stores the version information in the
25 memory. The document processing method further comprises a step of generating consistency-processed shared information already subjected to a consistency processing operation. The consistency processing step,

when the shared information is generated and a character string or structure has been already changed after editing of the document, maintains contents of the character string in its state before the change and
5 returns the structure change to its state at the time of creating the document, generates shared information satisfying its consistency, and, when it is desired to display and/or print the document in the output step, embeds the consistency-satisfied shared information in
10 the document.

In accordance with a still further aspect of the present invention, the document processing method further comprises a step of managing a change history of the shared information and document, the version
15 management step stores in the memory the generated shared information and generated document attached by information indicative of a new version, creates version information for each of the shared information and document and stores the version information in the
20 memory. The document processing method further comprises a step of generating consistency-processed shared information already subjected to a consistency processing operation. The consistency processing step, when the shared information is already edited after
25 editing of the document, acquires the latest shared information and the shared information at the time of creating the document to be displayed or printed, compares the acquired both shared information to extract

a construction changed location, and, when the structure change is extracted, returns the changed structure to its state at the time of creating the document in the latest shared information to generate consistency-
5 satisfied shared information, and, when it is desired to display or print the document in the output step, embeds the consistency-satisfied shared information in the document.

In accordance with an additional aspect of
10 the present invention, in the document processing method, when the structure is changed in the consistency processing step, the construction is deleted from the latest shared information when contents of the change corresponds to insertion of the structure in the shared
15 information at the document creation time, or, when the contents of the change corresponds to deletion of the structure from the shared information at the document creation time, the structure is inserted in the latest shared information, or, when the contents of the change
20 corresponds to exchange of the structure of the shared information at the document creation time, the structure after the exchange is deleted from the latest shared information and the structure before the exchange is inserted to thereby generate the consistency-satisfied
25 shared information.

Other objects, features and advantages of the present invention will become apparent when reading the following detailed description of the embodiments of the

invention taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows an arrangement of a first
5 embodiment of the present invention;

Fig. 2 is a processing flowchart for editing shared information in the first embodiment of the present invention;

Fig. 3 is a processing flowchart for editing a
10 structured document in the first embodiment of the present invention;

Fig. 4 is a processing flowchart for displaying or printing the structured document in the first embodiment of the present invention;

Fig. 5 shows an example for explaining a prior art sharing method of document parts in documents;

Fig. 6 shows a first example of shared information when represented by graphical user interface (GUI) function of a shared information editing program;

Fig. 7 shows an example when a first example of the shared information is output as a structured document;

Fig. 8 shows diagrams for explaining how the first example of the structured document is edited and
25 how the shared information is inserted to explain the first embodiment;

Fig. 9 is an example when a first example of

the structured document is output as a structured document;

Fig. 10 is an example when the first example of the structured document having the shared information embedded therein is displayed in the form of a layout;

Fig. 11 is a second example of the shared information when displayed by the GUI function of the shared information editing program;

Fig. 12 is an example when a second example of the shared information is output as a structured document;

Fig. 13 is an example when a second example of the structured document is output as a structured document;

Fig. 14 is an example when the second example of the structured document having the shared information embedded therein is displayed in the form of a layout;

Fig. 15 shows an arrangement of a second embodiment of the present invention;

Fig. 16 is a processing flowchart for editing shared information in the second embodiment of the present invention;

Fig. 17 is a processing flowchart for editing a structured document in the second embodiment of the present invention;

Fig. 18 is a processing flowchart for displaying or printing the structured document in the second embodiment of the present invention;

Fig. 19 shows examples of V1.0 and V2.0 after and before a second example of the shared information is edited;

Fig. 20 shows examples of version information
5 about edited structured document;

Fig. 21 shows an example of V2.0 after a second example of the structured document was edited;

Fig. 22 shows examples of version information about the edited structured document;

10 Fig. 23 shows an arrangement of a third embodiment of the present invention;

Fig. 24 is a processing flowchart for displaying or printing a structured document in the third embodiment of the present invention;

15 Fig. 25 is a processing flowchart for performing consistency processing operations over shared information in the third embodiment of the present invention;

20 Fig. 26 shows examples of V1.0 and V2.0 after and before a third example of the shared information is edited; -

Fig. 27 is an example when a third example of the structured document is output as a structured document;

25 Fig. 28 is an example when the third example of the structured document having the shared information embedded therein is displayed in the form of a layout;

Fig. 29 is an example when the third example

of the structured document having the shared information embedded therein is displayed in the form of a layout in the second embodiment;

Fig. 30 is an example when a third example of the shared information is differentially extracted;

Fig. 31 shows an example of the shared information already subjected to the consistency processing operation; and

Fig. 32 is an example when the third example of the structured document having the shared information already subjected to the consistency processing operation is displayed in the form of a layout.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention will be explained in connection with embodiments of the present invention. Although explanation will be made in the present embodiments in connection with a structured document having a document structure prepared using SGML by way of example, it should be noted that the present invention may be applied to general documents as well.

<Embodiment 1>

Fig. 1 shows an arrangement of a first embodiment of the invention. As illustrated, the present embodiment includes a CPU 101, a terminal unit 102 including input/output devices, a secondary memory 103 for storing documents therein, a memory medium 104 for storing programs therein, and a primary memory 1 for

storing a procedure program for processing various structured documents.

The structured document processing program further includes a shared information editing program 5 105 for editing shared information 103a frequently described in a plurality of structured documents, a shared information storage program 106 for storing the shared information 103a in the secondary memory 103, a shared information listing up program 107 for displaying 10 the stored shared information 103a for each of the types of the information, a structured document editing program 108 for editing a structured document 103b to describe a link of the shared information 103a in the edited structured document, a structured document 15 storage program 109 for storing the structured document 103b in the secondary memory 103, and a structured document output program 110 for embedding the contents of the shared information 103a in the structured document 103b according to link information described in 20 the document structured document 103b to generate information indicative of a structured document as an editing objective and to output the information in the form of a display or printout. An output of the structured document output program 110, that is, the 25 shared-information embedded document information can be provided to various types of user application devices. For example, the information can be provided not only to the aforementioned display or printout device but also

to a file means, network terminal or server. As an alternative to the hardware configuration in the above arrangement, a plurality of terminal units 102 may be connected to the CPU 101, primary memories 1, 103a and 5 103b, secondary memory 103, shared information 103a and structured document 103b through a bus or a communication network communication network 120.

In the present embodiment, an SGML document is used as the structured document as an example. SGML is 10 a document description language which is prescribed as a marked-up structured document based on 8879 of the International Organization for Unification (ISO). However, the present embodiment is not limited to SGML. For example, the present invention can be applied to a 15 structured document language such as XML and non-structured document languages with modifications. The shared information 103a and structured document 103b are stored in a memory such as a hard disk drive.

Explanation will next be made as to specific 20 processing procedures of the present embodiment by referring to such flowcharts as shown in Figs. 2, 3 and 4, followed by description of processing examples according to the processing procedures.

As shown in Fig. 2, editing operation of 25 shared information is carried out according to a procedure which follows.

Step 201:

The shared information editing program 105

receives many pieces of shared information alone frequently appearing in a plurality of structured documents, edits the received shared information, and stores it in the secondary memory 103 as the shared
5 information 103a.

At this time, information such as assumed to be later shared among a plurality of documents are selected as the shared information, described and modified according to the type of the information.

10 Step 202:

The edited shared information 103a is stored in the secondary memory 103 by execution of the shared information storage program 106.

Shown in the present embodiment is such an
15 example that shared information are stored as structured documents depending on the types of term information such as abbreviation or coinage and of person information such as person's name or belonging department. However, when it is unnecessary to store the shared
20 information as the structured document, the shared information may be stored by using a database management system or may be stored in an ordinary file system as data files.

As shown in Fig. 3, next, a structured
25 document is edited independently of the steps 201 and 202 in accordance with a procedure which follows.

Step 301:

The structured document editing program 108

performs structured document editing operation.

Step 302:

When it is desired to insert shared information in a structured document being edited, the shared
5 information list-up program 107 lists up the shared
information stored in the secondary memory 103.

At this time, an operator or user specifies
the type such as term or person of the shared
information to be inserted, which causes the program 107
10 to produce a list of the information for each of the
types.

Sep 303:

A link of the shared information to be
inserted is described to a target structured document by
15 execution of the structured document editing program
108.

The link description may be performed by means
of the user who directly writes a link representation
from an input device using the structured document
20 program 108, or also may be realized by means of the
user who gives the link representation through such
user's operation as a drag and drop of a selected term
in an insertion location of the objective structured
document using the shared information list-up program
25 107.

Step 304:

Completion of the editing operation of the
structured document causes the structured document

storage program 109 to store the document already edited in the secondary memory 103.

The structured document stored in the step 304 is displayed or printed in accordance with a procedure
5 which follows, as shown in Fig. 4.

Step 401:

When it is desired to display or print the structured document, the structured document to be displayed or printed as well as the shared information
10 about a link destination described in the structured document are read into the structured document output program 110 from the secondary memory 103.

Step 402:

The contents of the shared information is
15 embedded in the structured document, displayed or printed by execution of the structured document output program 110.

Processing Example 1:

The embodiment will be detailed in connection
20 with specific processing examples of Figs. 6 to 10 by referring to flowcharts of Figs. 2, 3 and 4.

Step 201:

The shared information editing program 105 receives many pieces of shared information alone
25 frequently appearing in a plurality of structured documents and edits the received shared information.

In the processing example, the shared information is assumed to be terms of character strings

of such company name or product name as to be commonly used throughout all the documents, or of such as module name which may be modified on the way in a project.

In the example, terms which are estimated to
5 be later shared in a plurality of documents as shared information, are selected and the shared information is described and modified depending on the types of sets of the terms.

For example, the shared information editing
10 program 105 has a graphical user interface (GUI) as shown in Fig. 6. In the present example, items of key and term are defined in term definition.

Of character strings to be described in documents, the user defines as sets of keys and terms
15 such character strings or terms that are desired to keep their expression consistencies or such terms that are desired to be used commonly throughout all the documents even when the terms are modified on the way. The key item is used when a link to the associated term is
20 described in a structured document, and is defined uniquely in all the terms.

Prepared in the GUI of Fig. 6 is a term definition which includes items of key and term and in which the user can define such a term such as a company
25 name or product name to be unified in all the documents or a module name of a project which may be modified on the way.

The GUI defines shared information of the

input terms. The shared information editing program 105 edits the shared information according to the definition to create a structured document as shown in Fig. 7.

The term shared information is described as
5 followed by a structure of <term definition> and as structured so as to have structures of <term> corresponding in number to the defined terms. Item key for each term describes the attribute of identification (ID) with respect to the construction <term>, and the actual
10 character string is expressed so as to indicate the contents of the structure.

Step 202:

The edited shared information is stored in the secondary memory 103 by execution of the shared
15 information storage program 106.

Upon storing the shared information, the term shared information is stored, for example, as such a structured document 701 as shown in Fig. 7.

A processing example of editing a structured
20 document is shown in the following.

Step 301:

The structured document editing program 108 edits a structured document.

For example, the structured document editing
25 program 108 has a GUI as shown in Fig. 8. In the drawing, a reference numeral 801 denotes a window for displaying therein a logical structure of a document, and a numeral 802 denotes a window for displaying

therein character strings in the logical structure.

Step 302:

When the shared information be inserted in the structured document now being edited, the shared
5 information list-up program 107 functions to display a list of the shared information of the terms stored in the secondary memory 103.

For example, the shared information list-up program 107 has a GUI similar to that of the shared
10 information editing program shared information editing program 105 shown in Fig. 6, and lists up terms defined as term shared information.

Step 303:

The structured document editing program 108
15 describes a link of a term to be inserted to the shared information.

In the example of Fig. 8, the user selects one of terms displayed by the shared information list-up program 107, drags and drops the selected term into an
20 insertion location to insert the term shared information in the structured document. This causes the structured document editing program 108 to describe its link. In this connection, a reference numeral 803 denotes a representation to be linked.

25 Further, the user can directly describe the link to the shared information from an input device by execution of the structured document editing program 108.

Fig. 9 shows an example of a structured document having links described therein. Such user's operation as shown in Fig. 8 causes generation of a structured document as shown in Fig. 9.

5 In the structured document of Fig. 9, the links are described in hypermedia/time-based structuring language (HyTime) (ISO/IEC 10744).

 In the SGML, for the purpose of expressing mutual reference between structures, an ID attribute is
10 given to one of the structures to be referenced while an identification (ID) of the referencing structure is given to the value of the IDREF attribute.

 In Fig. 9, a construction 902 of <TERM> is inserted as a link to a term and the term as a link
15 destination is specified in the IDREF attribute of the TERM. However, since direct IDREF reference to the ID described in another document cannot be made in the SGML, the HyTime language is used to reference the ID of the shared information in another document in this
20 example.

 As shown in Fig. 9, first, in an item 901, ENTITY reference to the structured document 701 of Fig. 7 having the shared information described therein is described in HyTime notation. The ENTITY reference is a
25 notation for referencing another document, following which, another structured document 701 of "term_dic.sgm" can be referenced under the ENTITY name of "term_dic".

 Descriptions by reference numerals 903 and 904

are such that a structure having the ID attribute can be referenced from inside of the document in

"term_dic.sgm". More specifically, the description 903 enables a "product_name" 905 as an ID in the

5 "term_dic.sgm" to be referenced with an ID of
"product_name_ref" 906, while the description 904
enables a "SGML_parser" 907 to be referenced with an ID
of "SGML_parser_ref" 908.

Since the "product_name_ref" 906 and
10 "SGML_parser_ref" 908 can make IDREF reference, these
are defined as attribute values of <TERM>, respectively.
Step 304:

Completion of editing operation of the
structured document causes the structured document
15 storage program 109 to store the edited document in the
secondary memory 103.

This means that such a structured document
shown in Fig. 9 is stored.

Explanation will next be made as to a
20 processing example of displaying or printing the
structured document of Fig. 9 stored in the step 34.

Step 401:

The structured document to be printed as well
as the shared information 701 of the link destination
25 term described in the structured document are read out
from the secondary memory 103.

Step 402:

The contents of the shared term information

structured document 701 is embedded in the structured document, and then displayed or printed by execution of the structured document output program 110.

Fig. 10 shows an example of displaying the structured document of Fig. 9. In the Figure, the contents of the shared term information described as links are embedded in the structured document and displayed in the form of a layout.

Through the above steps, links of terms frequently appearing in the structured documents to the shared information can be easily described in the structured documents.

As a result, the consistency of a term to be desirably used as the same term in different documents can be easily kept and thus efficient document editing can be realized. This is valid, in particular, when it is desired for a plurality of persons to edit term-consistent documents.

Further, even when a term is modified, only modification of its shared information causes automatic reflection of the modification on all the documents, thus realizing efficient document modification.

Processing Example 2:

Explanation will then be made of the second processing example of the first embodiment in accordance with flowcharts of Figs. 2, 3 and 4 with use of examples of Figs. 11 to 14.

Step 201:

The shared information editing program 105 edits shared information frequently appearing in a plurality of structured documents.

In this processing example 2, it is assumed
5 that the shared information is person information which includes person's name, belonging department and extension telephone number as a combination of terms of such character strings. It goes without saying that information other than the person information may be
10 employed as the combination terms.

In this example, information which is estimated as will be later shared in the plurality of documents is selected as the shared information, and described and modified for each of the types of sets of
15 the terms.

For example, the shared information editing program shared information editing program 105 has a GUI as shown in Fig. 11. In this Figure, person information include items of key, name, belonging department and
20 extension telephone number.

The user defines person information to be described in the documents as a set of person's key, name, belonging department and extension telephone number. Item key is to be used when a link to the
25 person information is described in the structured document, and is defined uniquely throughout all person information.

Prepared in the GUI of Fig. 11 is a term

definition which includes of name, belonging department and extension telephone number of person information.

Using the term definition, the user can define the person information to be unified throughout all the

5 documents.

The GUI defines shared information of input person information. This causes the shared information editing program 105 to edit the shared information according to the definition to thereby create a

10 structured document shown in Fig. 12 as an example of person dictionary file.

The shared information of the present example is described as structured in such a manner that a construction of <person information> is followed by

15 constructions of <person> corresponding in number to defined persons. Further, a key for each person is described as an ID attribute of a structure <person>. Each person information has such a structure as to have a structure of <person> followed by structures of the

20 person's <name>, <belonging department> and <extension telephone number>, which are described as the contents of their structures.

Step 202:

Shared information 1101 of the edited person

25 information is stored in the secondary memory 103 by execution of the shared information storage program 106.

Upon storing the shared information, the shared information is stored in the memory 103b, for

example, as a structured document 1201 as shown in Fig. 12.

The following is an processing example of editing a structured document.

5 Step 301:

The structured document editing program 108 edits a structured document. The structured document editing program 108 has a GUI similar to that of Fig. 8.
Step 302:

10 When the shared information of the person information is inserted in the structured document being edited, the shared person information stored in the secondary memory 103 is listed up by execution of the shared information list-up program 107.

15 For example, the shared information list-up program 107 has a GUI similar to that of the shared information editing program 105 shown in Fig. 11, and lists up the defined person information.

Step 303:

20 A link to the shared information to be inserted is described by execution of the structured document editing program 108. The link description is carried out in the same manner as in the processing example 1.

25 Shown in Fig. 13 is an example of a structured document having links already described therein. In the Figure, similarly to Fig. 9, the HyTime language is used to describe the links to the shared information.

As shown in Fig. 13, first of all, ENTITY reference to the structured document 1201 of Fig. 12 having the shared information described therein is described in a representation 1301 in the HyTime notation. Due to such description, another structured document 1201 referred to herein as "person_dic.sgm" can be referenced by the ENTITY name of "person_dic".

In a description 1304, a structure of "taro" 1202 as an ID in the "person_dic.sgm" can be referenced with use of an ID of "person_ref" 1309.

In a description 1305, an ID of "person_name_ref" 1310 is described in the 'treeloc' notation of the HyTime language to indicate the first child structure of its own as viewed from the structure of "person_ref".

More in detail, the description 1305 has a construction <name> having contents of "Heisei Taro". The first "1" as the contents of <marklist> given below <treeloc> means its own and the next "1" means the first child.

In a description 1306, an ID of "person_belong_ref" 1311 indicates a structure of the second child as viewed from a structure of "person_ref". In a description 1307, an ID of "person_tel_ref" 1312 indicates a construction of the third child as viewed from the construction of "person_ref".

The respective ID's, since the ID's can receive the IDREF reference from inside of the document,

are defined as attribute values of <TERM>, respectively.

Further, when the "taro" 1308 is changed to "jiro", its name, belonging department and extension telephone number indicate a child of a construction having an ID "jiro". Thus, a "jiro" report can be created without any need for modifying the <TERM> description 1302 in Fig. 13 and "treeloc" descriptions 1305, 1306 and 1307.

Step 304:

10 Completion of the editing of the structured document causes the structured document storage program 109 to store the edited document in the secondary memory 103.

That is, such a structured document as shown in Fig. 13 is stored in the memory 103.

Next shown is a processing example of displaying or printing the structured document of Fig. 13 stored in the step 304.

Step 401:

20 A structured document to be displayed or printed as well as shared document 1201 of the link destination described in the structured document are read out from the secondary memory 103 into the memory 110.

25 Step 402:

By execution of the structured document output program 110, the contents of the shared person information 1201 is embedded in the structured document,

and then output in the form of a display or printout.

Fig. 14 shows an example of displaying the structured document of Fig. 13. In the Figure, the contents of the shared person information described as
5 links, that is, having its name "Heisei Taro", belonging department "development depart. 12 group" and extension telephone number "1111" are embedded in the structured document and displayed in the form of a layout.

Through the above steps, similarly to the
10 processing example 1, links to the shared information of a combination of terms frequently used in the documents can be easily described in the structured documents. Thus, even when person's belonging department or extension telephone number is modified, only modifi-
15 cation of the shared information of the combination terms causes automatic reflection of the modification contents on all the documents, thereby enabling automatic and efficient modification of the documents.

Further, when the HyTime notation is used, a
20 link can be described as a relative position from its KEY. Hence, once one document is previously prepared as a template, only modification of KEY ID enables automatic modification of all the associated information, thus increasing the document preparation
25 efficiency.

Furthermore, when the shared information of combination terms are previously saved as a structured document, modification of a structure of the shared

combination term information can be facilitated. For example, upon adding persons's addresses to the person information, the structure of the structured document of this Figure is converted by an SGML converting tool so that it is only required to insert a structure <address> below the structure <person>. Therefore, this management of the structure modification of the shared combination term information can be made easier than the schema modification when a database is used.

10 <Embodiment 2>

An arrangement of a second embodiment is shown in Fig. 15.

As illustrated, the present embodiment includes, in addition of the arrangement (see Fig. 1) of the first embodiment, a version management program 1501 which, the shared information 103a and structured document 103b are to be stored, stores them as a new version without overwriting them to manage a modification history of document creation date, time, etc. When a structured document is to be displayed or printed by execution of the structured document output program 110, the version management program 1501 extracts the shared information of latest one of versions of documents prepared prior to the creation date and time by the program structured document output program 110, embeds the shared information in the structured document, and then displays or prints the information-embedded document.

In other words, in accordance with the present embodiment, even when the shared information is modified, the modification is not reflected on past documents already edited, and the past shared information at the editing time is embedded for display or printout.

For example, when a report already submitted in the past is later displayed or printed, it becomes necessary to output the report without changing the reporter's name and belonging department described at the time of the report submission.

In the case of such a term used in a project, on the contrary, it becomes necessary to reflect the latest revision information even on the documents already edited.

In order to manage the consistency between the above two cases, whether to employ either one of the two cases is previously determined for each of the types of the document and shared information, and is previously stored as document attribute information.

The processing operation at the time of displaying or printing a document is changed by referring to the attribute information, so that the above two-way consistency management can be realized.

When the shared information modification is to be reflected on the shared documents to display or print the structured documents having the shared information embedded therein, this can be realized by embedding the

latest version of shared information in the documents.

The operation of the present embodiment will be explained in connection with the case where the shared information modification is not to be reflected
5 on the shared documents.

A specific processing procedure of the present embodiment will be explained with reference to flowcharts of Figs. 16 to 18, and then processing examples will be explained according to the processing
10 procedure.

In Fig. 16 showing a procedure of editing shared information, a step 201 is the same as that in the processing procedure of the first embodiment.

Step 1601:

15 When the shared information storage program 106 is executed to store the shared information 103a edited in the step 201 in the secondary memory 103, control is passed to the version management program 1501, and the shared information is stored as a new
20 version one without overwriting by execution of the program-version management program 1501.

The version management program 1501 is executed to store in the secondary memory 103 version information 103c including version numbers and creation
25 date and time of all version shared information already edited, associate the version information 103c with all the already-edited version shared information, and then store the edited version shared information in the

secondary memory 103.

As a result, by specifying any version number, shared information on the specified version number can be later extracted.

5 Next, editing operation of a structured document is carried out according to the following procedure independently of the steps 201 and 1601.

In Fig. 17 showing a processing procedure, steps 301 to 303 are the same as those in the processing
10 procedure of the first embodiment.

Step 1701:

When the structured document 103b edited in the step 303 is stored in the secondary memory 103 by execution of the structured document storage program
15 109, the document 103 is stored without overwriting it as a new version document by execution of the version management program 1501.

The version management program 1501 stores in the secondary memory 103 the version information 103c
20 including version numbers and creation date and time of all versions of edited documents, associates the version information 103c with all the versions of edited documents, and then stores the edited version document in the secondary memory 103.

25 As a result, by specifying a desired version number, a structured document of the desired version can be later extracted.

Further, when it is desired to display or

print the structured document 103b, a procedure is carried out according to the flowchart of Fig. 18.

In the operations of this Figure, the contents of latest one of versions of the shared information prepared prior to the creation date and time of the structured document to be displayed or printed is embedded in the structured document, and then the embedded document is displayed or printed.

Step 1801:

10 The creation date and time of the structured document to be displayed or printed is acquired by execution of the version management program 1501, and is set as TIME 1.

Step 1802:

15 The version information 103c of the shared information to be embedded in the structured document is acquired by execution of the version management program 1501.

Step 1803:

20 The creation date and time of the latest version of the shared information is acquired by execution of the version management program 1501. This creation date and time is set as TIME 2.

Step 1804:

25 When TIME 2 is earlier than TIME 1, the processing proceeds to a step 1805. When TIME 2 is not earlier than TIME1, the shared information is of the version created after the time point of having described

the links in the structured document, the creation date and time of a one-previous version is acquired. This creation date and time is replaced by TIME 2 and the processing returns to the step 1804.

5 Step 1805:

When TIME 2 is earlier than TIME 1, the shared information is information at the time at which the links have been described in the structured document, and thus the version of shared information is acquired.

10 Step 1806:

By execution of the structured document output program 110, the content of the shared information of the version acquired in the step 1805 is embedded in the structured document, and the embedded document is

15 displayed or printed.

(Processing Example 3):

A specific processing example of the present embodiment will be explained with use of the document of Figs. 19 to 22 with reference to the flowcharts of Figs.

20 17 and 18.

Step 1601:

When the shared information edited in the step 201 is stored in the secondary memory 103 by execution of the shared information storage program 106, the
25 edited shared information is stored as a new version by execution of the version management program 1501.

It is assumed that, for example, since the belonging department of a person "Heisei Taro" has been

changed, the shared information 1101 as the person
information of Fig. 11 has been edited as shown in Fig.
19. In this case, the shared information 1101 of the
person information shown in Fig. 11 is set as version
5 V1.0, while shared information shared information 1901
of the person information after edited is set as version
V2.0. By execution of the version management program
1501, version information 2001 of creation date and time
as shown in Fig. 20 is stored.

10 The following is a processing example of
editing a structured document.

Step 1701:

When the structured document edited in the
step 303 is stored in the secondary memory 103 by
15 execution of the structured document storage program
109, the edited structured document is stored as a new
version by execution of the version management program
1501.

For example, it is assumed that the structured
20 document 1401 of Fig. 14 has been edited as another
report as shown in Fig. 21 and that the structured
documents of the old and new versions as well as version
information 2201 of creation date and time, etc. as
shown in Fig. 22 have been stored.

25 As shown in Fig. 22, the structured document
1401 of V1.0 is created on "1996.5.15,9:00", and the
shared information inserted at that time is of V1.0.
The shared information is edited on "1996.8.21,15:30" as

shown in Fig. 20 and stored as a new version V2.0. Accordingly, since the structured document 2101 of V2.0 is created on "1996.11.20,16:30", the shared information inserted at that time is of V2.0.

5 Further, explanation will be then made of an example of displaying or printing the structured documents of Figs. 14 and 21, by referring to the flowchart of Fig. 18.

Step 1801:

10 The creation date and time of the structured document to be displayed or printed is acquired by execution of the version management program 1501. The creation date/time is set as TIME 1.

 In the case of the structured document 1401 of
15 Fig. 14, "1996.5.15,9:00" is substituted for TIME 1.

 In the case of the structured document 2101 of Fig. 21, "1996.5.11,16:30" is substituted for TIME 1.

Step 1802:

 The version information of the shared
20 information to be embedded in the structured document is acquired by execution of the version management program 1501.

 The version information 2001 of the shared information of Fig. 20 is acquired.

25 Step 1803:

 The creation date and time of the latest version of the shared information is acquired by execution of the version management program 1501. The

creation date and time is set as TIME 2. Here,
"1996.8.21, 15:30" is substituted for TIME 2.

Step 1804:

When TIME 2 is not earlier than TIME 1, this
5 means that the shared information is edited after the
time point at which the link has been described in the
structured document. Thus, the creation date and time
of the one-previous version is acquired. The acquired
creation date and time is substituted for TIME 2, and
10 then, the processing returns to the step 1804.

In the case of the structured document 1401 of
Fig. 14, since TIME 2 is later than TIME 1, the creation
date and time "1996.4.1,10:00" of the one-previous
version V1.0 is substituted for TIME 2. At this time
15 point, TIME 2 is earlier than TIME 1.

In the case of the structured document 2101 of
Fig. 21, TIME 2 is earlier than TIME 1.

Step 1805:

When TIME 2 is earlier than TIME 1, the shared
20 information is the information at the time point at
which the link has been described in the structured
document, so that the shared information of that version
is acquired.

In the case of the structured document 1401 of
25 Fig. 14, the version V1.0 is acquired.

In the case of the structured document 2101 of
Fig. 21, the version V2.0 is acquired.

Step 1806:

The content of the shared information acquired in the step 1805 is embedded in the structured document and displayed or printed by execution of the structured document output program 110.

5 In the case of the structured document 1401 of Fig. 14, the content of the shared information of V1.0 is embedded therein, so that its belonging department column has "development department 12 group" corresponding to the content at the time of creating the
10 structured document.

 In the case of the structured document 2101 of Fig. 21, the content of the shared information of V2.0 is embedded therein, so that its belonging department column has "development department 16 group" corresponding to the content at the time of creating the
15 structured document.

 Through the above steps, when a structured document created in the past is displayed or printed, the shared information at the time point of having
20 created the structured document can be embedded therein.

 It will be easily appreciated that, when a structured document is copied and edited to create another document, it is unnecessary to change the description of the link to the shared information and
25 the shared information at that time point is automatically embedded, thus facilitating document re-use.

 When shared information to be embedded in a structured document mixedly contains such shared

information as person information requiring the
aforementioned version selection and shared information
such as term information for which the latest version is
always selected; a flag is provided at each link
5 location in the structured document. In the case of the
person information, the flag is set; while, in the case
of the term information, the flag is not set. When the
flag is set in the display or print mode, operations as
shown by the flowchart in Fig. 18 are carried out to
10 select a version and to embed the shared information.
When the flag is not set, the latest version is selected
to embed the shared information.

In addition to combination terms such as
person information indicative of a combination of term,
15 person's name and belonging department as shown in
Embodiments 1 and 2, there are various types of shared
information which are valid when the shared information
are stored according to their types, listed up and
shared between structured documents.

20 When HyTime is used, a link to image or audio
data can also be described. Thus, when personal
photographic information (image data) or brief self
introduction by her or his speech (audio data) as an
example is previously stored in association with a
25 person, the present invention can be applied to various
types of documents including writer introduction of
papers or web pages in world wide web (WWW).

In accordance with the present invention, such

photographic information is previously stored as
upgraded at suitable intervals, so that, when a past
document such as a paper is displayed or printed, the
information at the time point of having created the
5 document can be used, and when an active document such
as web pages is displayed or printed, the latest
information can be used, thus realizing more flexible
information re-use.

<Embodiment 3>

10 In Embodiment 2, the structured document has
been correctly displayed or printed, when the shared
information is edited but it is desired not to reflect
the modification on past documents already edited and
having links to the shared information, by embedding the
15 shared information at the time of having created the
structured document.

Embodiment 3 is intended, when shared
information is edited but it is desired to reflect the
modification on past documents already edited and having
20 links to the shared information, to correctly display or
print a structured document even when the link
correlation varies due to the influence of the
modification of the shared information.

Explanation will first be made of a problem
25 that the link correlation varies.

It is now assumed that shared information 2601
of V1.0 has been edited to create a shared information
2602 of V2.0 as shown in Fig. 26, for instance.

Fig. 26 shows shared information of persons. The shared information of V2.0 contains, in addition to the information of V1.0, structures 2603 and 2604 of <name number>. In addition, the belonging department of
5 "Heisei Taro" and extension telephone number as a character string are changed.

Assume that, prior to the editing of the shared information of Fig. 26 to V2.0, a structured document 2700 having a link to the shared information
10 2601 described therein as shown in Fig. 27 has already existed.

In the structured document 2700, similarly to the structured document of Fig. 13 shown by the processing example 2 of Embodiment 1, the link is
15 described based on HyTime notation as a relative position with respect to KEY.

In a description 2701, first of all, ENTITY reference to the shared information 2601 of Fig. 26 stored under a file name of "person_dic.sgm" is
20 described. Due to this description, the document shared information 2601 of "person_dic.sgm" can be referenced with the ENTITY name of "person_dic".

In a description 2704, a structure "taro" as an ID in "person_dic.sgm" can be referenced with use of
25 an ID of "person_ref" 2709.

In a description 2705, an ID of "person_name_ref" 2710 indicates the first child of its own as viewed from a structure "person_ref". That is,

it indicates a structure <name> having the contents of "Heisei Taro". The first "1" in the contents of <marklist> given below <treeloc> means its own, and the next "1" means the first child.

5 In a description 2706, similarly, an ID of "person_belong_ref" 2711 indicates a structure of the second child as viewed from a structure "person_tel_ref" 2712 indicates a structure of the third child as viewed from a structure of "person_ref".

10 The respective IDs, which can be subjected to the IDREF reference from inside the document, are defined as attribute values of <TERM>, respectively.

 When the structured document described in such a manner as mentioned above is displayed by execution of
15 the structured document output program 110 in accordance with the steps 401 and 402; the contents of the shared information described as the links, i.e., the name "Heisei Taro", belonging department "development department 12 group" and extension telephone number "1111" are
20 embedded in the structured document, and the embedded structured document is displayed in the form of a layout as shown in Fig. 28.

 When the structured document 2700 is displayed or printed after the shared information 2601 is edited
25 into V2.0 information, the system shown in Embodiment 1 involves a problem described below.

 That is, the structured document 2700 is a document for describing self introduction therein, so

that when it is desired to display or print the document, the latest information should be always displayed or printed. To this end, when the latest shared information 2602 is displayed as embedded in the structured document, it is undesirably displayed as shown in Fig. 29, because the link is described as a relative position with respect to KEY.

This means that, though the belonging department is defined in the structured document 2700 as the second child from the viewpoint of having an ID "taro", insertion of a new structure in the shared information of V2.0 results in the belonging department not being the second child, thus changing its link correlation.

In Fig. 29, the contents of <name number> as the second child of a structure having an ID "taro" is displayed as the contents of the belonging department in the latest shared information 2602. Similarly, representation of its extension telephone number is also shifted and is not correct.

It is desirable that a link destination originally refers to the same structure as that at the time of creating it and contents thereof be the latest ones.

The above problem can be avoided by describing the link in the form of not the relative position but an absolute position using a structure ID. However, this system also eliminates such a merit as exemplified in

Embodiments 1 and 2, that is, only KEY modification enables automatic modification of all related information.

In order to embed the contents of the latest shared information in the structured document having the links to the shared information described therein and to prevent a structure discrepancy, the present embodiment provides such an arrangement as shown in Fig. 23.

As illustrated, the present embodiment is arranged, in addition to the arrangement (see Fig. 15) of Embodiment 2, so that, when shared information is edited after creation of a structured document to thereby change a character string or structure of the shared information, a consistency processing program 2301 performs its consistency processing operation on the shared information to create new shared information already subjected to the consistency processing operation, whereby the consistency-processed shared information can be displayed or printed as embedded in the structured document by execution of the structured document output program 110.

The consistency-processed shared information generated by the consistency processing program 2301 refers to such information that, with respect to shared information after a character string or structure has been changed, the contents of the character string are the same as those after the change, the structure change causes the structured document to be returned to its

creation time state, so that, by embedding it in the structured document, the change in the contents of the character string are reflected and the original link correlation can be correctly maintained.

5 A specific processing procedure of the present embodiment will be explained with use of the flowcharts of Figs. 24 and 25.

 After that, a processing example will be explained in accordance with the processing procedure.

10 The editing procedure of shared information and the editing procedure of a structured document are the same as the operations of the steps 201 and 202 and steps 301 to 304 indicated in Embodiment 1. The structured document is displayed in accordance with a
15 processing procedure, as shown in Fig. 24, described below.

Step 2401:

 A structured document to be displayed or printed is acquired from the secondary memory 103 by
20 execution of the version management program 1501.

Step 2402:

 When the structured document with the latest version shared information embedded therein is displayed or printed, consistency operation is carried out to the
25 latest shared information by execution of consistency processing program 2301 to generate shared information subjected to the consistency operation. The details will be later explained in steps 2501 to 2504. When the

latest version shared information is not embedded, a step 2404 is performed.

Step 2403:

5 The contents of the consistency-processed shared information generated in the step 2402 are embedded in locations described as links in the structured document to obtain an embedded structured document, and then the embedded structured document is displayed or printed

10 Step 2404:

 When the latest version shared information is not to be embedded in the structured document for its display or printout, the shared information at the time of creating the structured document is acquired by
15 execution of the version management program 1501 in accordance with a procedure similar to that of the steps 1801 to 1805 in Embodiment 2.

Step 2405:

 The contents of the shared information
20 acquired in the step 2404 are embedded at locations described as links in the structured document, and then the structured document is displayed or printed by execution of the structured document output program 110.

 Detailed explanation will be made of a
25 processing procedure of subjecting the latest shared information to the consistency operation to generate consistency-processed shared information by execution of the consistency processing program consistency

processing program 2301, indicated in the step 2402 by referring to the flowchart of Fig. 25.

Step 2501:

The latest shared information is acquired by
5 execution of the version management program 1501.

Step 2502:

Shared information at the time of creating the
structured document is acquired by execution of the
version management program 1501 in accordance with a
10 procedure similar to that of the steps 1801 to 1805
shown in Embodiment 2.

Step 2503:

The latest shared information acquired in the
step 2501 is compared with the shared information at the
15 time of creating the structured document acquired in the
step 2502 to extract a structure-changed location.

Step 2504:

When the structure-changed location is
extracted in the step 2503, this means that there may be
20 a discrepancy in the link to the shared information. In
this case, the structure change is returned to its state
at the time of creating the structured document for its
modification to thereby generate consistency-processed
shared information.

25 First, in order to return the structure change
to the state at the document creation time, if a new
structure is added to the shared information at the
document creation time, then the new additional

structure is deleted from the latest shared information.

On the contrary, if a structure in the shared information at the document creation time is deleted from the new shared information, then the creation-time
5 structure is added to the latest shared information. The structure exchange is regarded as structure deletion and insertion, which is carried out in the same manner as the above.

Such modified latest shared information is
10 called shared information which has been modified so as to keep the consistency.

(Processing Example 4)

A specific processing example of the embodiment will be explained in connection with, as an
15 example, the shared information of Fig. 26 and the structured document of Fig. 27 with reference to the flowcharts of Figs. 24 and 25.

Step 2401:

A structured document to be displayed or
20 printed is acquired from the secondary memory 103 by execution of the version management program 1501. In this case, the structured document of Fig. 27 is acquired.

Step 2402:

25 When the structured document with the latest version shared information embedded therein is to be displayed or printed, the latest shared information is consistency-processed to generate consistency-processed

shared information.

A processing example of the step 2402 will be detailed with use of the flowchart of Fig. 25.

Step 2501:

5 The latest shared information is acquired by execution of the version management program 1501. The shared information 2602 of the latest V2.0 is acquired in Fig. 26.

Step 2502:

10 The shared information at the time of creating the structured document is acquired by execution of the version management program 1501 according to a procedure similar to that of the steps 1801 to 1805 shown in Embodiment 2.

15 Since the structured document 2700 of Fig. 27 has been already created before the editing of the shared information V2.0, the shared information 2601 of V1.0 is acquired.

Step 2503:

20 The latest shared information acquired in the step 2501 is compared with the shared information at the document creation time acquired in the step 2502 to extract a structure changed location.

25 There are considered several methods of extracting a structure changed location. Employed in the present processing example, however, is, as an example, a difference extracting method by execution of a program which compares the old and new structured

documents.

Fig. 30 shows a difference extracting result between the shared information of V1.0 and V2.0 acquired in the steps 2501 and 2052.

Fig. 30 shows that extraction is made such that character string changes are made at locations 3003 and 3004 shown by underlines in Fig. 30, and structure changes are made at locations 3005 and 3006 shown by areas shaded by slanted lines.

10 Step 2504:

When a structure changed location is extracted in the step 2503, there may be a discrepancy in the link to the shared information. To avoid this, the structure change in the latest shared information is returned and modified to the state at the time of having created the structured document to thereby generate consistency-processed shared information.

It will be seen from Fig. 30 that the structures 3005 and 3006 are newly added in the latest shared information. In order to return the latest information to the state at the document creation time, these structures are deleted from the latest shared information. In this connection, as a difference extraction technique, there can be employed a technique disclosed in U.S. Serial No. 08/657,306, by Aoyama et al., assigned to the same assignee as the present application. The disclosure of that application is incorporated herein by reference.

Fig. 30 shows the consistency-processed shared information modified in this way. In Fig. 31, the newly-added structures <name number> are deleted, but the character string changed locations remain as they are as the latest contents 3102.

Step 2403:

The contents of the consistency-processed shared information generated in the step 2402, i.e., steps 2501 to 2504 are embedded at locations described as links in the structured document, and the embedded structured document is displayed or printed by execution of the structured document output program 110.

Fig. 32 shows an example of the consistency-processed shared information of Fig. 31 when the structured document of Fig. 27 is embedded and displayed. As shown in Fig. 32, no discrepancy takes place in the structures described as link destinations at the document creation time while the contents of the shared information are displayed as the latest information in the form of a layout.

When the latest shared information is not to be embedded, the operations of the steps 2404 and 2405 are carried out, which are the same as those in Embodiment 2 and thus explanation thereof is omitted.

Through the above steps, the shared information is edited. When the modification change be reflected on structured documents commonly having the shared information, the latest contents can be reflected

on the documents while preventing any shift of the link correlation between the documents and shared information.

The functional programs of the memories shown
5 in Embodiments of Figs. 1, 15 and 23 can be stored in a memory medium such as CD-ROM, DVD-ROM or floppy disk to be installed from the medium to an implementation device. Alternatively, these programs may also be installed from a server into the primary memory 1.
10 These programs may be installed in various aspects, as will be appreciated easily.